UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/749,079	12/30/2003	Vincent J. Zimmer	42.P18117	7951

7590 10/16/2007

R. Alan Burnett
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
Seventh Floor
12400 Wilshire Boulevard
Los Angeles, CA 90025-1026

EXAMINER	
VICARY, KEITH E	
ART UNIT	**PAPER NUMBER**
2183	
MAIL DATE	**DELIVERY MODE**
10/16/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/749,079	**Applicant(s)** ZIMMER ET AL.	
	Examiner Keith Vicary	**Art Unit** 2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *30 December 2003*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-30* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *30 December 2003* is/are: a)☒ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

1. Claims 1-30 are pending in this office action and presented for examination.

Specification

2. The disclosure is objected to because of the following informalities. Appropriate correction is required.

a. In paragraph [0023], second to last line, "only a set 106 up exception vectors" should presumably be "only a set 106 of exception vectors."

Claim Objections

3. Claims 12 and 29 are objected to because of the following informalities. Appropriate correction is required.

b. Claims 12 and 29 recite the limitation "configured to handler" in line 5 which should presumably be "configured to handle."

Claim Rejections - 35 USC § 112

4. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

5. Claims 8-9 and 13-17 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

6. Claim 8 recites the limitation "the base address of the second memory address space" in lines 8-9. There is insufficient antecedent basis for this limitation in the claim.

c. Claim 9 is rejected for failing to alleviate the rejection of claim 8 above.

7. Claim 13 recites the limitation “physically or logically replacing the set of OS-based exception handler components” in lines 4-6. It is indefinite as to what constitutes physical replacement as this could mean that the system memory itself is replaced or alternatively that the entries within the system memory are replaced. Similiarly, it is indefinite as to logical replacement entails the changing of pointers or the changing of data entries with the actual physical memory remaining the same.

d. Claims 14-17 are rejected for failing to alleviate the rejection of claim 13 above.

Claim Rejections - 35 USC § 101

8. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

9. Claims 18-25 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

e. A claim to a proper computer readable medium encoded with functional descriptive material must effect a useful, concrete, and tangible result.

f. Functional descriptive material must be claimed in combination with an appropriate tangible computer readable medium in order to be statutory.

10. Claims 18 and 22 are not claimed in combination with an appropriate tangible computer readable medium. In view of Applicant's disclosure, specification page 24, paragraph [0075], the machine-readable medium is not limited to tangible embodiments, instead being defined as including both tangible embodiments (e.g. recordable media, such as disk-based media) and intangible embodiments (e.g., propagated signals such as electrical, optical, acoustical, or other form of propagated signals...carrier waves, infrared signals, digital signals, etc). As such, the claim is not limited to statutory subject matter and is therefore non-statutory.

g. Claims 19-21 and 23-25 are rejected for failing to alleviate the rejection of claims 18 and 22 above.

11. To alleviate this rejection, the examiner recommends amending the claims to clearly specify a "machine readable disk-based medium" instead of a mere "machine-readable medium."

Claim Rejections - 35 USC § 103

12. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

13. Claims 1-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Branch (WO 01/093022 A3) in view of Krishnaswamy (US 6735774 B1).

14. Consider claim 1, Branch discloses vectoring an instruction pointer to a firmware-based exception filter in response to an exception (page 1, paragraph A, updating the appropriate pointer in the interrupt vector table so that it points to the new interrupt service routine; this causes the program counter to then point to the new interrupt service routine; an interrupt is a type of exception; note that paragraph B discloses of device drivers, which are firmware based); executing the firmware-based exception filter (page 1, paragraph A, chaining a new interrupt service routine; paragraph B, the new interrupt service routine performs an operation); and re-vectoring the instruction pointer to an exception handler configured to handle the exception (page 1, paragraph A, instruction at the end of the new interrupt service routine that causes program flow to jump to the address of the existing interrupt service routine; this address that enters the program counter is the instruction pointer to the OS exception handler).

However, Branch does not explicitly disclose that the exception handler is an *operating system* exception handler, as Branch does not limit his existing exception handler to a specific type.

On the other hand, Krishnaswamy does disclose of operation system exception handlers (for example, col. 2, lines 27-30, system calls through a vector table).

Supporting system calls allows for the use of functions such as forking, profiling, and tracing which increases functionality (Krishnaswamy, col. 1, lines 10-15; Table 1 functions).

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Krishnaswamy with the invention of Branch in order to increase functionality. It would have been readily recognized to one of ordinary skill in the art at the time of the invention that the environment of Krishnaswamy is analogous to the environment of Branch: Krishnaswamy's overall invention of having a system vector table supplemented by an alternative vector table correlates to Branch's existing and new interrupt service routines. It is noted that a system call is essentially a type of interrupt itself.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Krishnaswamy with the invention of Branch in order to increase functionality.

15. Consider claim 2, Krishnaswamy discloses execution of the firmware-based exception filter performs operations including saving at least one processor register value to a storage device (col. 2, lines 52-56, dump counter values).

16. Consider claim 3, Krishnaswamy discloses execution of the firmware-based exception filter performs operations including saving at least a portion of system memory to a storage device (col. 2, lines 52-56, dump counter values).

17. Consider claim 4, Branch discloses loading a set of OS exception handler pointers into a first memory address space; relocating the set of OS exception handler pointers to a second memory address space; and loading a set of firmware-based exception filter pointers into the first address space (page 1, paragraph A, updating the

appropriate pointer in the interrupt vector table so that it points to the new interrupt service routine; note that the program flow eventually jumps to the address of the existing interrupt service routine, thus the pointer to that existing interrupt service routine must be relocated somewhere).

18. Consider claim 5, Branch discloses storing a base address of the second memory address space; and employing the base address of the second memory address space to re-vector the instruction pointer to an OS exception handler pointer to the OS exception handler configured to handle the exception (page 1, paragraph A, updating the appropriate pointer in the interrupt vector table so that it points to the new interrupt service routine; note that the program flow eventually jumps to the address of the existing interrupt service routine, thus the pointer to that existing interrupt service routine must be relocated somewhere). Note for reference that Krishnaswamy discloses in, for example, Figure 2, of updating the system vector instead of the program counter.

19. Consider claim 6, Branch discloses loading a set of OS exception handlers into a first memory address space; relocating the set of OS exception handlers to a second memory address space; and loading a set of firmware-based exception filters into the first address space (page 1, paragraph A, difficult to store the new interrupt service routine when the interrupt vector table contains actual code for the interrupt service routines; even though Branch is saying it is difficult, he is nevertheless disclosing it).

20. Consider claim 7, Branch discloses storing a base address of the second memory address space and employing the base address of the second memory address space to re- vector the instruction pointer to the OS exception handler configured to handle the exception (page 1, paragraph A, updating the appropriate pointer in the interrupt vector table so that it points to the new interrupt service routine; note that the program flow eventually jumps to the address of the existing interrupt service routine, thus the pointer to that existing interrupt service routine must be relocated somewhere; this is applied to Branch's disclosure of the vector table containing actual code for the interrupt service routines). Note for reference that Krishnaswamy discloses in, for example, Figure 2, of updating the system vector instead of the program counter.

21. Consider claim 8, Branch discloses loading a set of OS exception handler pointers into a first memory address space; setting a processor exception vector register to include a base address of the first memory address space; loading a set of firmware-based exception filter pointers into a second address space; and replacing the base address of the first memory address space with the base address of the second memory address space in the processor exception vector register (page 1, paragraph A, updating the appropriate pointer in the interrupt vector table so that it points to the new interrupt service routine; note that the program flow eventually jumps to the address of the existing interrupt service routine, thus the pointer to that existing interrupt service

routine must be relocated somewhere; a program counter is read broadly to be the processor exception vector register; note the first and second address space may be the same). Note for reference that Krishnaswamy discloses in, for example, Figure 2, of updating the system vector instead of the program counter.

22. Consider claim 9, Branch discloses storing a base address of the first memory address space; and employing the base address of the first memory address space to re-vector the instruction pointer to an OS exception handler pointer to the OS exception handler configured to handle the exception (page 1, paragraph A, updating the appropriate pointer in the interrupt vector table so that it points to the new interrupt service routine; note that the program flow eventually jumps to the address of the existing interrupt service routine, thus the pointer to that existing interrupt service routine must be relocated somewhere). Note for reference that Krishnaswamy discloses in, for example, Figure 2, of updating the system vector instead of the program counter.

23. Consider claim 10, Branch discloses loading a set of OS exception handlers into a first memory address space; setting a processor exception vector register to include a base address of the first memory address space; loading a set of firmware-based exception filters into a second address space; and resetting the processor exception vector register to include a base address of the second memory address space (page 1, paragraph A, difficult to store the new interrupt service routine when the interrupt vector

table contains actual code for the interrupt service routines; even though Branch is saying it is difficult, he is nevertheless disclosing it).

24. Consider claim 11, Branch discloses storing a base address of the first memory address space; and employing the base address of the first memory address space to re-vector the instruction pointer to the OS exception handler configured to handle the exception (page 1, paragraph A, updating the appropriate pointer in the interrupt vector table so that it points to the new interrupt service routine; note that the program flow eventually jumps to the address of the existing interrupt service routine, thus the pointer to that existing interrupt service routine must be relocated somewhere; this is applied to Branch's disclosure of the vector table containing actual code for the interrupt service routines). Note for reference that Krishnaswamy discloses in, for example, Figure 2, of updating the system vector instead of the program counter.

25. Consider claim 12, Branch discloses loading the firmware-based exception filter into system memory (page 1, paragraph A, store the new interrupt service routine in the interrupt vector table; alternatively, it is inherent that a pointer in an interrupt vector table points somewhere that the system can access, thus the exception filter is in system memory); and fixing up code in the firmware-based exception filter to re-vector the instruction pointer to one of the OS exception handler configured to handle the exception or a pointer to the OS exception handler configured to handler the exception (page 1, paragraph A, including an instruction at the end of the new interrupt service

routine that causes program flow to jump to the address of the existing interrupt service routine).

26. Consider claim 13, Branch discloses loading a set of exception handler components into system memory (page 1, paragraph A, updating the appropriate pointer in the interrupt vector table; for an update to occur, an original set must have been loaded previously; an interrupt is a type of exception; note that paragraph B discloses of device drivers, which are firmware based); physically or logically replacing the set of exception handler components with a corresponding set of firmware-based exception filter and/or handler components (page 1, paragraph A, updating the appropriate pointer in the interrupt vector table so that it points to the new interrupt service routine); vectoring an instruction pointer to a firmware-based exception filter and/or handler in response to an OS runtime exception (page 1, paragraph b, the new interrupt service routine performs an operation immediately before the existing interrupt service routine) ; and executing the firmware-based exception filter and/or handler (page 1, paragraph b, the new interrupt service routine performs an operation immediately before the existing interrupt service routine).

However, Branch does not explicitly disclose that the exception handler is an *operating system* exception handler, as Branch does not limit his existing exception handler to a specific type.

On the other hand, Krishnaswamy does disclose of operation system exception handlers (for example, col. 2, lines 27-30, system calls through a vector table).

Supporting system calls allows for the use of functions such as forking, profiling, and tracing which increases functionality (Krishnaswamy, col. 1, lines 10-15; Table 1 functions).

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Krishnaswamy with the invention of Branch in order to increase functionality. It would have been readily recognized to one of ordinary skill in the art at the time of the invention that the environment of Krishnaswamy is analogous to the environment of Branch: Krishnaswamy's overall invention of having a system vector table supplemented by an alternative vector table correlates to Branch's existing and new interrupt service routines. It is noted that a system call is essentially a type of interrupt itself.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Krishnaswamy with the invention of Branch in order to increase functionality.

27. Consider claim 14, Branch discloses re-vectoring the instruction pointer to an operating system (OS) exception handler configured to handle the OS run-time exception after the firmware-based exception filter and/or handler has been executed (page 1, paragraph b, the new interrupt service routine performs an operation immediately before the existing interrupt service routine).

28. Consider claim 15, Branch discloses fixing up code in the firmware- based exception filter and/or handler to re-vector the instruction pointer to one of the OS exception handler configured to handle the OS runtime exception or a pointer to the OS exception handler configured to handle the OS runtime exception (see the rejection of claim 12).

29. Consider claim 16, Branch discloses the set of OS-based exception handlers are physically replaced by: copying the set of OS-based exception handlers from a physical address space to a virtual address space; and overwriting the physical address space with the set of firmware-based exception filter and/or handler components (see the rejection of claim 6).

30. Consider claim 17, Branch discloses the set of OS-based exception handlers are logically replaced by: loading the set of OS-based exception handlers into a first memory address space having a first base address; and loading the set of firmware-based exception filter and/or handler components into a second address space having a second base address; and replacing the first base address with the second base address in a register that is used to locate the base address of a table containing one of a set of exception handler procedures or pointers to a set of exception handler procedures (see the rejection of claim 10).

31. Consider claim 18, Branch discloses determining a first base address of a set of exception handler components that have been loaded into a first memory address space; storing the first base address (page 1, paragraph a, interrupt vector table is at a fixed location); loading a set of firmware-based exception filter and/or handler components into a second memory address space having a second base address; and setting an exception vector register to have a base address corresponding to the second base address (page 1, paragraph A, difficult to store the new interrupt service routine when the interrupt vector table contains actual code for the interrupt service routines; even though Branch is saying it is difficult, he is nevertheless disclosing it). Note for reference that Krishnaswamy discloses in, for example, Figure 2, of updating the system vector instead of the program counter.

However, Branch does not explicitly disclose that the exception handler is an *operating system* exception handler, as Branch does not limit his existing exception handler to a specific type.

On the other hand, Krishnaswamy does disclose of operation system exception handlers (for example, col. 2, lines 27-30, system calls through a vector table).

Supporting system calls allows for the use of functions such as forking, profiling, and tracing which increases functionality (Krishnaswamy, col. 1, lines 10-15; Table 1 functions).

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Krishnaswamy with the invention of Branch in order to increase functionality. It would have been readily recognized to one of ordinary skill in the art at the time of the invention that the environment of Krishnaswamy is analogous to the environment of Branch: Krishnaswamy's overall invention of having a system vector table supplemented by an alternative vector table correlates to Branch's existing and new interrupt service routines. It is noted that a system call is essentially a type of interrupt itself.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Krishnaswamy with the invention of Branch in order to increase functionality.

32. Consider claim 19, Branch discloses the set of firmware-based exception filter and/or handler components (page 6, paragraphs D and E, computer readable memory containing both a device driver and the instructions to chain the interrupt service routines).

33. Consider claim 20, Branch discloses the medium comprises a firmware storage device (page 6, paragraphs D and E, computer readable memory containing both a device driver and the instructions to chain the interrupt service routines).

34. Consider claim 21, Branch discloses filtering a runtime exception using a firmware-based exception filter; and re-vectoring an instruction pointer to an operating system (OS) exception handler configured to handle the runtime exception (page 1, paragraph B, combine a new interrupt service routine with an existing interrupt service

routine so that the new interrupt service routine performs an operation immediately before the existing interrupt service routine; paragraph A, jump to the address of the existing interrupt service routine).

35. Consider claim 22, Branch discloses moving a set of exception handler components from a first memory address space having a first base address to a second memory address space having a second base address; storing the second base address; and loading a set of firmware-based exception filter and/or handler components into the first memory address space (page 1, paragraph A, updating the appropriate pointer in the interrupt vector table so that it points to the new interrupt service routine; note that the program flow eventually jumps to the address of the existing interrupt service routine, thus the pointer to that existing interrupt service routine must be relocated somewhere; this is applied to Branch's disclosure of the vector table containing actual code for the interrupt service routines).

36. Consider claim 23, see the rejection of claim 19.

37. Consider claim 24, see the rejection of claim 20.

38. Consider claim 25, see the rejection of claim 21.

39. Claims 26-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Branch in view of Krishnaswamy and in further view of Brannock (US 20020194313).

40. Consider claim 26, Branch discloses a processor (page 1, paragraph F, microprocessor); memory, coupled to the processor (page 1, paragraph G, memory); a device, having firmware instructions stored thereon to perform operations in combination with logic programmed into the processor (page 1, paragraph B, new interrupt service routine), the operations including: loading a firmware-based exception filter into memory (page 1, paragraph A, updating the appropriate pointer in the interrupt vector table so that it points to the new interrupt service routine; thus the new interrupt service routine must inherently have been loaded into memory); detecting a runtime exception (it is inherent that a runtime exception is detected for any interrupt chaining to occur); vectoring an instruction pointer to the firmware-based exception filter in response to the runtime exception (page 1, paragraph A, the program counter will then point to the firmware-based exception filter); executing the firmware-based exception filter; and re-vectoring the instruction pointer to an exception handler configured to handle the runtime exception (page 1, paragraph B, the new interrupt service routine performs an operation immediately before the existing interrupt service routine; paragraph A, jump to the address of the existing interrupt service routine).

However, Branch does not explicitly disclose that the exception handler is an *operating system* exception handler, as Branch does not limit his existing exception handler to a specific type.

On the other hand, Krishnaswamy does disclose of operation system exception handlers (for example, col. 2, lines 27-30, system calls through a vector table).

Supporting system calls allows for the use of functions such as forking, profiling, and tracing which increases functionality (Krishnaswamy, col. 1, lines 10-15; Table 1 functions).

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Krishnaswamy with the invention of Branch in order to increase functionality. It would have been readily recognized to one of ordinary skill in the art at the time of the invention that the environment of Krishnaswamy is analogous to the environment of Branch: Krishnaswamy's overall invention of having a system vector table supplemented by an alternative vector table correlates to Branch's existing and new interrupt service routines. It is noted that a system call is essentially a type of interrupt itself.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Krishnaswamy with the invention of Branch in order to increase functionality.

However, neither Branch nor Krishnaswamy disclose that the device is a flash device.

On the other hand, Brannock discloses of a flash device ([0022], flash ROM part).

It would have been obvious to one of ordinary skill in the art at the time of the invention for the device to be a flash device as the implementation of the flash device as the device is a simple substitution of one known element for another to obtain predictable results.

41. Consider claim 27, Brannock discloses of a network interface coupled to the processor, wherein execution of firmware instructions loads a firmware-based exception filter from a network storage device via the network interface into the memory ([0022], remote directory on a server). It would have been obvious to one of ordinary skill in the art at the time of the invention to combine this further teaching of Brannock with the previously combined invention of Branch, Krishnaswamy, and Brannock, in order to prevent error from improperly installing new BIOS chips ([Brannock, [0006]).

42. Consider claim 28, see the rejection of claim 10.

43. Consider claim 29, see the rejection of claim 12.

44. Consider claim 30, see the rejection of claim 6.

Conclusion

45. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

h. Harrington et al. (US 20050027972 A1) discloses of sharing an exception vector between firmware and an operating system.

i. Mealey et al. (US 5758168) discloses of interrupt vectoring for optionally architected facilities in computer systems. He discloses of using architecture specific firmware in order to perform routines such as performance monitoring and is particularly relevant to the applicant's disclosure.

j. Srivastava et al. (US 7194744 B2) discloses dynamic exception handling using an external exception handler and is particularly relevant due to his disclosure of both a local exception handler and an external exception handler.

46. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Keith Vicary whose telephone number is (571) 270-1314. The examiner can normally be reached on Monday - Friday, 8:00 a.m. - 5:00 p.m., EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on 571-272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

RICHARD L. ELLIS
PRIMARY EXAMINER

kv